



TECHNISCHE UNIVERSITÄT
CHEMNITZ

R Crashkurs

Professur Finanzmathematik, Fakultät für Mathematik

R Crashkurs

Dr. Dana Uhlig
Professur Finanzmathematik
Fakultät für Mathematik

Umfrage: Erfahrungen mit R

- A – keine
- B – reingeschnuppert, ich kann einfache statistische Grafiken, Kennzahlen und Tests berechnen
- C – Umgang mit R bekannt, Umsetzung statistischer Verfahren neu
- D – Vollprofi

Onlineabstimmung <http://twbk.de>

Aktuelles Thema

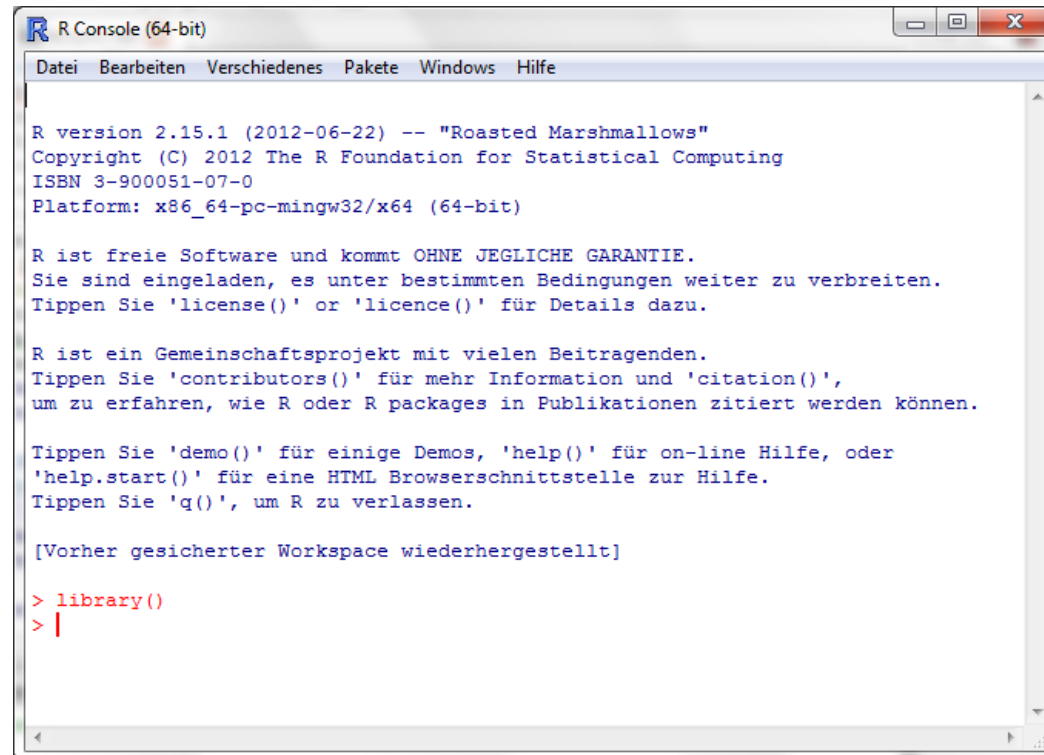
1. Erste Schritte in R
2. Datenstrukturen in R
3. Schritt für Schritt mehr R
4. Erste Statistiken mit R

Was ist R?

- ▶ Statistiksoftware / Programmiersprache, aktuelle Version 3.5.0 (2018-04-23) basierend auf der Sprache R
- ▶ freie Software (open source software, GNU General Public License):
`http://www.r-project.org/`
- ▶ verfügbar für viele Plattformen (Windows, Linux, Mac, Unix, ...)
- ▶ Entwicklung: 1992 von Ross Ihaka und Robert Gentleman an der Universität Auckland in Anlehnung an kommerzielle Software S
- ▶ flexible Programmiersprache
- ▶ Interpretersprache: Benutzereingaben in der Kommandozeilenkonsole ⇒
Betätigung Enter-Taste ⇒ unmittelbare Ausführung
- ▶ Ausführung von Skripten (Anreihung mehrerer Kommandos)
- ▶ zahlreiche Pakete zur Erweiterung des Funktionsumfangs

Arbeiten mit R, I

► R-Standard-Konsole



```
R R Console (64-bit)
Datei Bearbeiten Verschiedenes Pakete Windows Hilfe

R version 2.15.1 (2012-06-22) -- "Roasted Marshmallows"
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-mingw32/x64 (64-bit)

R ist freie Software und kommt OHNE JEDLICHE GARANTIE.
Sie sind eingeladen, es unter bestimmten Bedingungen weiter zu verbreiten.
Tippen Sie 'license()' or 'licence()' für Details dazu.

R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information und 'citation()',
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.

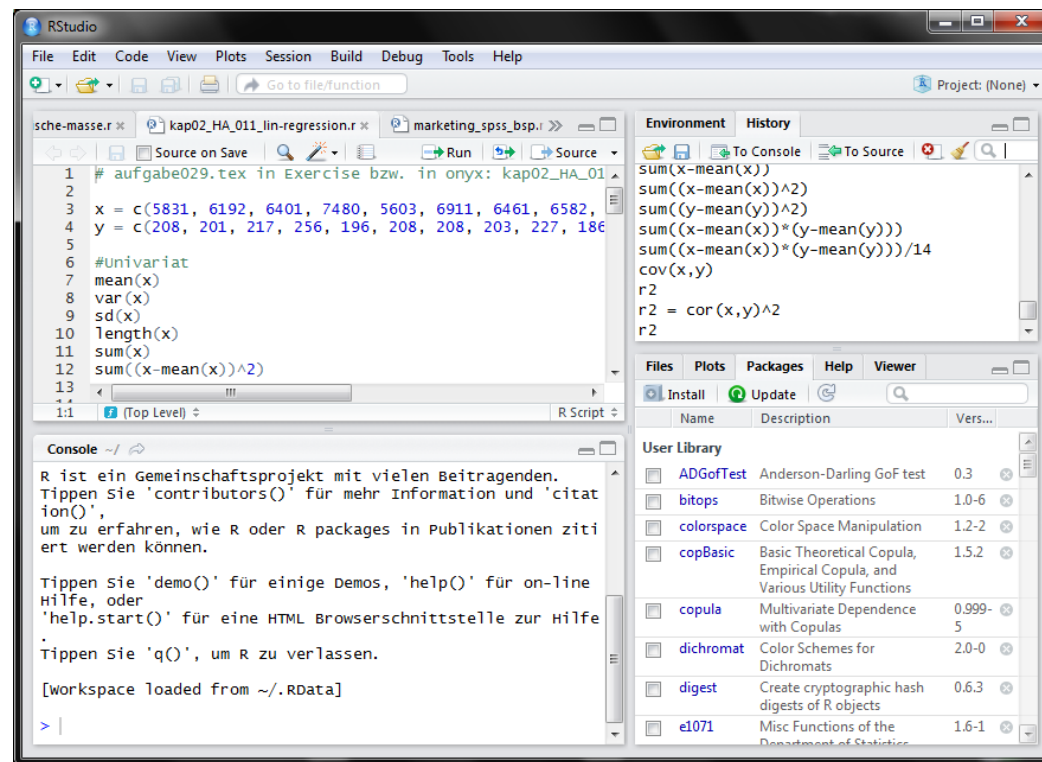
Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.

[Vorher gesicherter Workspace wiederhergestellt]

> library()
> |
```

Arbeiten mit R, II

- ▶ hilfreiche Editoren: zahlreiche grafische Benutzeroberflächen (z.B. R Commander, JGR (Jaguar), RStudio, Tinn-R, ...)
- ▶ hier in der Lehrveranstaltung: RStudio, da unter Windows, Linux und Mac frei verfügbar <http://www.rstudio.com/>



R als einfacher Taschenrechner

```
5 + 2  
## [1] 7  
  
3 * 6  
## [1] 18
```

Arbeiten mit Variablen

```
a = 5  
b = 2  
(c=sqrt(a^2+b^2))  
## [1] 5.385165
```

Konventionen

```
#Kommentare werden vom Interpreter ignoriert

#1. Zuweisungen mittels = oder <-
a = 5
b <- 4
c = a + b
c

## [1] 9

x = seq(from = 1, to = 10, by = 0.2)
x[1:5]

## [1] 1.0 1.2 1.4 1.6 1.8

#Aufrufen von Funktionen
y = log(x)

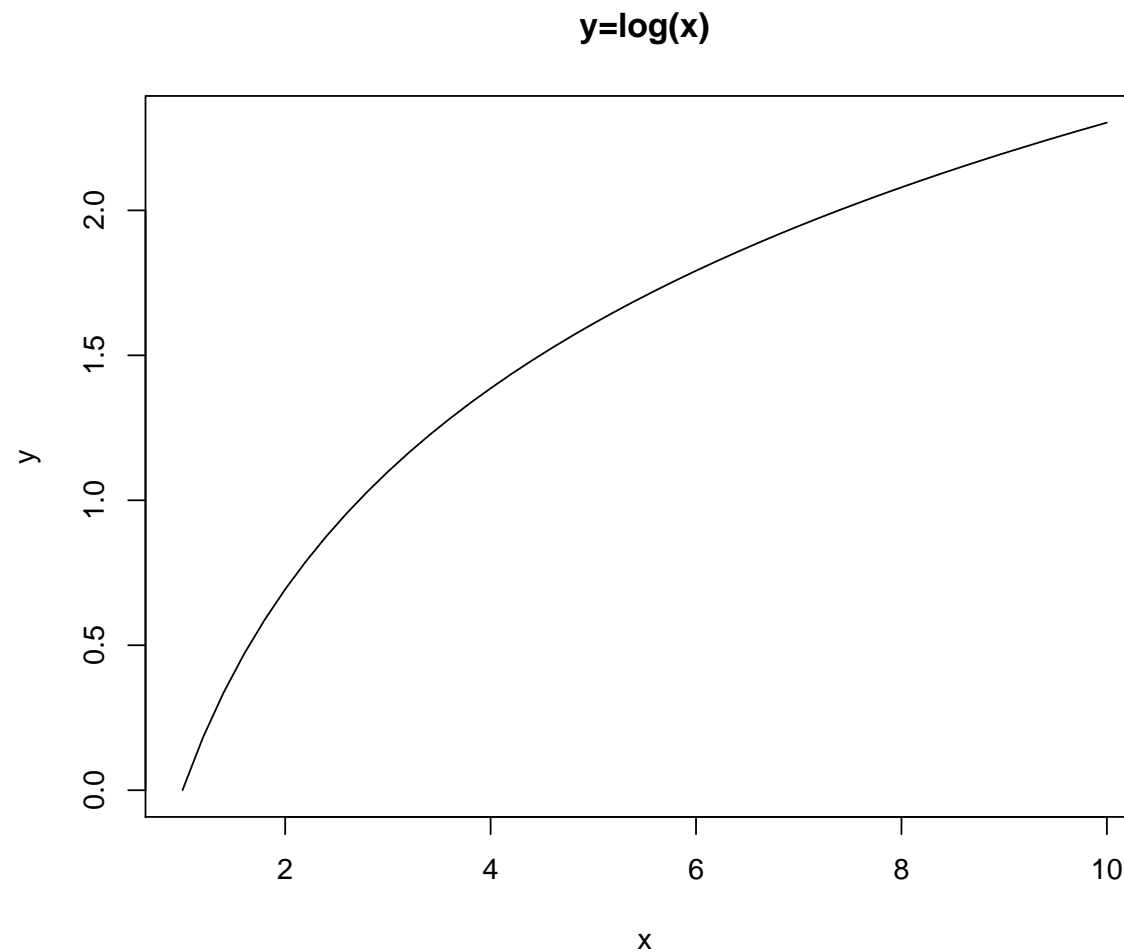
#die ersten 5 Werte von y
y[1:5]

## [1] 0.0000000 0.1823216 0.3364722 0.4700036 0.5877867
```


Konventionen

```
#einfache Grafiken
```

```
plot(x, y, type = "l", main = "y=log(x) ")
```



Erste Statistik mit einem simulierten Datensatz

```
set.seed(1121)
x=rnorm(n=200, mean=100, sd=2)
#die ersten 8 Werte von x
x[1:8]

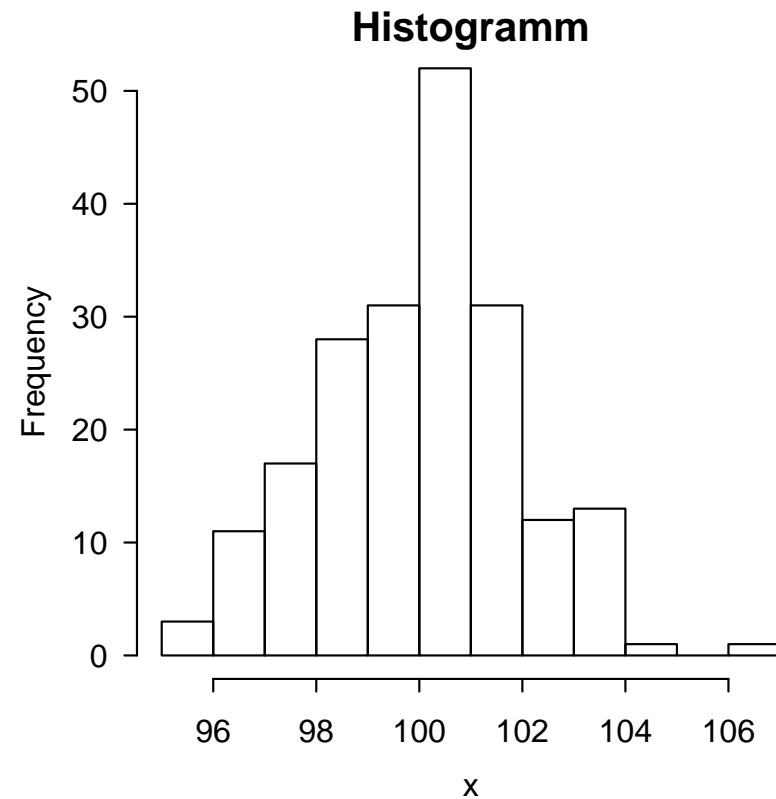
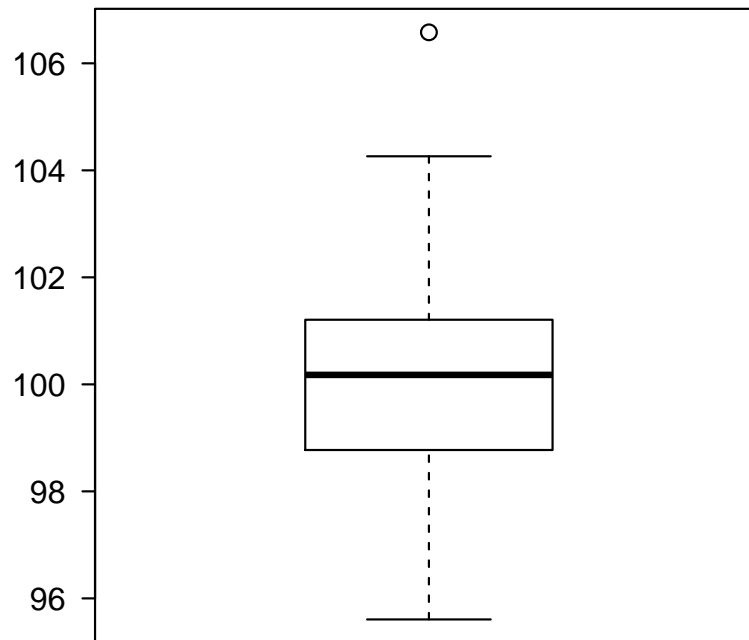
## [1] 100.28992 100.87664 100.30638 102.16989 103.99909 98.37623 100.32054 100.04639

mean(x); var(x)

## [1] 100.0464
## [1] 3.774443
```

Das erste Element von x ist 100.2899166 und der empirische Mittelwert $\bar{x} = 100.0463889$ und die empirische Varianz $s^2 = 3.7744426$.

```
boxplot(x)  
hist(x, main='Histogramm')
```





Funktionserweiterung mittels Paketen

Zusatzpakete einmalig installieren und vor Gebrauch in R-Session laden

```
#einfache loesung: packet einfach installieren mittels Aufruf
#install.packages("quantmod")

#eleganter: zuerst abfragen, ob Pakete bereits installiert sind
wants = c("quantmod", "mvtnorm", "nortest", "foreign")
has     = wants %in% rownames(installed.packages())
if(any(!has)) install.packages(wants[!has], repos = "https://cloud.r-project.org",
                               lib=.libPaths()[1])

#Pakete laden
library("quantmod")

## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric
## Loading required package: TTR
## version 0.4-0 included new data defaults. See ?getSymbols
```

Zahlreiche Datensätze bereits in R

```
head(mtcars, n = 6)
```

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.00	6.00	160.00	110.00	3.90	2.62
Mazda RX4 Wag	21.00	6.00	160.00	110.00	3.90	2.88
Datsun 710	22.80	4.00	108.00	93.00	3.85	2.32
Hornet 4 Drive	21.40	6.00	258.00	110.00	3.08	3.21
Hornet Sportabout	18.70	8.00	360.00	175.00	3.15	3.44
Valiant	18.10	6.00	225.00	105.00	2.76	3.46

↪ für Übungszwecke und zur Veranschaulichung statistischer Verfahren

Arbeitsverzeichnis = Working Directory

```
# aktuelles Arbeitsverzeichnis abfragen
```

```
getwd()
```

```
## [1] "/home/mathe/lvhome50/users/personal/dana/svn/Angewandt"
```

```
#mittels setwd("PFAD-NEUES-VERZEICHNIS")
```

```
#neues Arbeitsverzeichnis festlegen
```

```
setwd()
```

- ▶ Alternative in Rstudio: Menü „Session“ → „Set working directory“
- ▶ wichtig bei Arbeiten mit Datenfiles oder für Aufruf weiterer Skripte (mittels `source()`)
- ▶ relative Pfade anstatt absolute

Aktuelles Thema

1. Erste Schritte in R

2. Datenstrukturen in R

Vektoren

Matrizen und Arrays

Data Frames

Listen

Faktoren

3. Schritt für Schritt mehr R

4. Erste Statistiken mit R

verschiedene Datenstrukturen und Datentypen aller Objekte in R, Abfrage in R mittels

```
mode() typeof() class() str()
```

▶ Datentypen

- ▶ *numeric* (Unterscheidung zwischen *integer* und *double*) ↷ Zahlen
- ▶ *complex* ↷ komplexe Zahlen
- ▶ *character* ↷ Zeichenkette / String
- ▶ *logical* ↷ logische Werte (TRUE bzw. FALSE)
- ▶ *raw* ↷ Bytes

▶ Klassen (komplexe Datentypen)

- ▶ Vektoren
- ▶ Faktoren
- ▶ Matrizen / Arrays
- ▶ Listen
- ▶ Data Frames
- ▶ Funktionen

Erstellung von Vektoren mittels `c()`

```
(a = c(5, 3, 2))  
  
## [1] 5 3 2  
  
(b = c("A", "B", "C"))  
  
## [1] "A" "B" "C"  
  
(x = 1:6)  
  
## [1] 1 2 3 4 5 6  
  
typeof(a) ; class(a)  #mode(a)  
  
## [1] "double"  
## [1] "numeric"  
  
is.numeric(b) ; is.character(b)  
  
## [1] FALSE  
## [1] TRUE
```

```
a  
  
## [1] 5 3 2  
  
#Laenge des Vektors  
length(a)  
  
## [1] 3  
  
#das 2. Element  
a[2]  
  
## [1] 3  
  
#das 2. bis 4. Element von x  
x[2:4]  
  
## [1] 2 3 4  
  
x+3  
  
## [1] 4 5 6 7 8 9
```

```
x*3

## [1] 3 6 9 12 15 18

(x = 1:6)

## [1] 1 2 3 4 5 6

#loeschen des 3. Wertes
(x = x[-3])

## [1] 1 2 4 5 6

#Anhängen weiterer Komponenten an einen bestehenden Vektor
(y1 = append(x, c(10,11), after=5))

## [1] 1 2 4 5 6 10 11

(y2 = c(-1, 0, x, 10))

## [1] -1 0 1 2 4 5 6 10
```

Vergleiche und Zugriff auf bestimmte Komponenten

```
a > 4

## [1] TRUE FALSE FALSE

a == 3

## [1] FALSE TRUE FALSE

#Wie viele Vektor-Komponenten sind groesser als 3
sum(a>3)

## [1] 1

#Welche Vektor-Komponenten sind groesser als 2.5 sowie deren Werte
which(a>2.5)

## [1] 1 2

a[a>2.5]

## [1] 5 3
```

Matrix \triangleq Anordnung mehrerer Vektoren in einer Tabelle , **Array** \triangleq beliebige Dimension

```
gewicht = c(56, 63, 80, 49, 75)
groesse = c(1.64, 1.73, 1.85, 1.6, 1.81)
alter = c(22, 21.5, 24, 28, 39)

#spaltenweise (c = columns) zusammenbinden
(S = cbind(alter, gewicht, groesse))

##      alter gewicht groesse
## [1,]  22.0      56    1.64
## [2,]  21.5      63    1.73
## [3,]  24.0      80    1.85
## [4,]  28.0      49    1.60
## [5,]  39.0      75    1.81

#zeilenweise (r = rows) zusammenbinden
(Z = rbind(alter, gewicht, groesse))

##      [,1] [,2] [,3] [,4] [,5]
## alter  22.00 21.50 24.00 28.0 39.00
## gewicht 56.00 63.00 80.00 49.0 75.00
## groesse  1.64  1.73  1.85  1.6  1.81
```

Zugriff auf Matrizen analog zu Vektoren: $S[i, j]$ entspricht dem Eintrag in der i -ten Zeile und j -ten Spalte von S

```
S[5, 1]

## alter
##      39

S[1, ]

##      alter gewicht groesse
##      22.00   56.00    1.64

S[c(1, 3, 5), ]

##           alter gewicht groesse
## [1, ]        22      56    1.64
## [2, ]        24      80    1.85
## [3, ]        39      75    1.81
```

Namen vergeben

```
colnames(S)

## [1] "alter" "gewicht" "groesse"

namen = c("Gerda", "Karin", "Hans", "Doris", "Ludwig")
rownames(S) = namen
S["Doris", "gewicht"]

## [1] 49

S[4, 2]

## [1] 49
```

hilfreiche Funktionen zum Arbeiten mit Matrizen oder Arrays

```
#Matrix-Dimension  
length(S)  
  
## [1] 15  
  
dim(S)  
  
## [1] 5 3  
  
nrow(S); ncol(S)  
  
## [1] 5  
## [1] 3  
  
#structure  
str(S)  
  
## num [1:5, 1:3] 22 21.5 24 28 39 56 63 80 49 75 ...  
## - attr(*, "dimnames")=List of 2  
## ..$ : chr [1:5] "Gerda" "Karin" "Hans" "Doris" ...  
## ..$ : chr [1:3] "alter" "gewicht" "groesse"
```


Alternative Erzeugung einer Matrix aus Vektor am Beispiel für $x = \begin{pmatrix} 1 \\ 2 \\ \vdots \\ 12 \end{pmatrix}$

```
x = 1:12
matrix(data = x, nrow=3)

##           [,1] [,2] [,3] [,4]
## [1,]         1   4   7  10
## [2,]         2   5   8  11
## [3,]         3   6   9  12
```

```
matrix(data = x, ncol=3)

##           [,1] [,2] [,3]
## [1,]         1   5   9
## [2,]         2   6  10
## [3,]         3   7  11
## [4,]         4   8  12
```

- ▶ häufigste Verwendungsform für statistische Daten
- ▶ matrixähnliche Struktur für verschiedene Datentypen
- ▶ Beispiel

```
S

##          alter gewicht groesse
## Gerda    22.0      56     1.64
## Karin    21.5      63     1.73
## Hans     24.0      80     1.85
## Doris    28.0      49     1.60
## Ludwig   39.0      75     1.81

sex = c("weiblich", "weiblich", "männlich", "weiblich", "männlich")
(daten = data.frame(S, sex))

##          alter gewicht groesse      sex
## Gerda    22.0      56     1.64 weiblich
## Karin    21.5      63     1.73 weiblich
## Hans     24.0      80     1.85 männlich
## Doris    28.0      49     1.60 weiblich
## Ludwig   39.0      75     1.81 männlich
```

Zugriff auf Data Frames

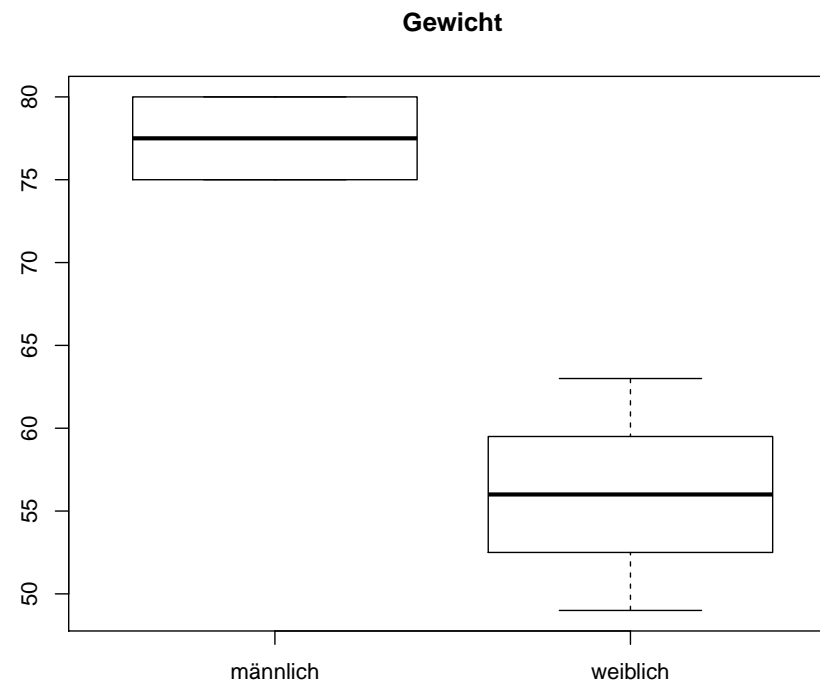
```
daten$gewicht  
  
## [1] 56 63 80 49 75  
  
daten[, "gewicht"]  
  
## [1] 56 63 80 49 75  
  
daten["Doris", ]  
  
##      alter gewicht groesse      sex  
## Doris    28      49     1.6 weiblich  
  
daten[4, ]  
  
##      alter gewicht groesse      sex  
## Doris    28      49     1.6 weiblich
```

Zugriff auf Data Frames

```
#direkter Zugriff auf Unterelemente mittels Suchpfad durch  
#attach() setzen  
attach(daten)  
  
## The following objects are masked _by_ .GlobalEnv:  
##  
##   alter, groesse, sex  
  
gewicht  
  
## [1] 56 63 80 49 75  
  
#Suchpfad fuer Objekt "daten" wieder loeschen  
detach(daten)  
#Zugriff auf Unterelemente wieder nur via objekt$unterobjekt  
daten$gewicht  
  
## [1] 56 63 80 49 75
```

Arbeiten mit Data Frames

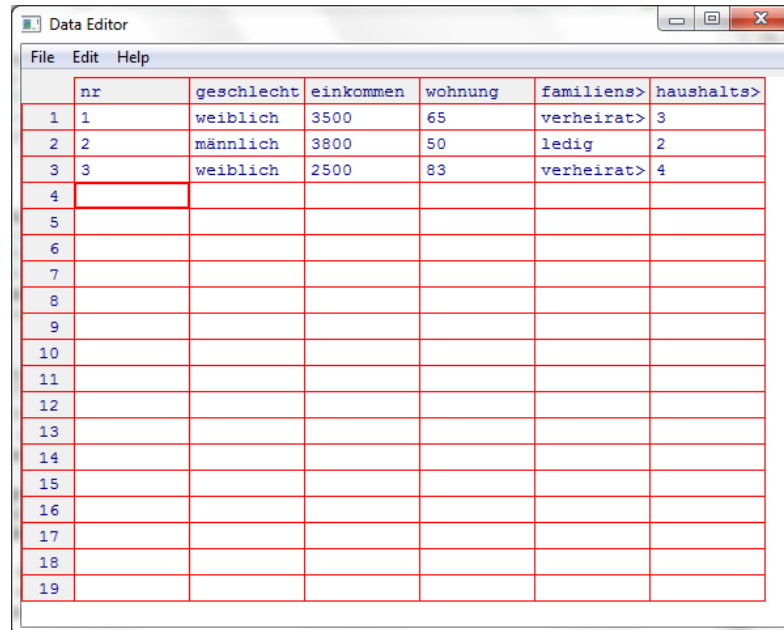
```
boxplot (daten$gewicht~daten$sex, main = "Gewicht")
```



alternative Dateneingabe

```

#1. Leeren Data Frame anlegen
mydata = data.frame()
#2. Beispielsweise Daten über Data Editor manuell eingeben
mydata = edit(mydata)
#weitere nützliche Befehle
head(mydata); fix(mydata); View(mydata);
save(mydata, file = "mydata.RData"); load("mydata.RData")
  
```



	nr	geschlecht	einkommen	wohnung	familiens>	haushalts>
1	1	weiblich	3500	65	verheirat>	3
2	2	männlich	3800	50	ledig	2
3	3	weiblich	2500	83	verheirat>	4
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						

- ▶ bei Vektoren und Matrizen sind alle Elemente vom gleichen Datentyp (schwächste)

```
a = c(2.0, 1, "a")
typeof(a)

## [1] "character"
```

- ▶ Listen zum Abspeichern beliebiger Datenstrukturen
- ▶ Details für rekursive Listen (geschachtelte Listen) siehe Literatur
- ▶ einfaches Beispiel

```
a = 1:5; b = c("a", "b", "c")
sex = as.factor(c("männlich", "weiblich"))
sim = rnorm(n=10, mean = 100, sd = 4)
mylist = list(a, b, sex, sim)
str(mylist)

## List of 4
## $ : int [1:5] 1 2 3 4 5
## $ : chr [1:3] "a" "b" "c"
## $ : Factor w/ 2 levels "männlich", "weiblich": 1 2
## $ : num [1:10] 106.4 100.6 97.8 101.3 89 ...
```

Warum Faktoren?

- ▶ nützlich für Umgang mit **kategorialen** Variablen
- ▶ Vektoren vom Typ *char* werden von R nicht als Faktor wahrgenommen
- ▶ **wichtig:** für ANOVA mittels *aov()* oder *lm()* muss Faktorvariable als Faktor definiert sein
- ▶ Kontrolle mittels *is.factor()*
- ▶ Umwandlung mittels *as.factor()*


```
a = c("ja", "nein")
str(a)

## chr [1:2] "ja" "nein"

is.factor(a)

## [1] FALSE

a.factor.1 = factor(a)
str(a.factor.1)

## Factor w/ 2 levels "ja","nein": 1 2

#gleiches Ergebnis liefert as.factor()
a.factor.2= as.factor(a)
str(a.factor.2)

## Factor w/ 2 levels "ja","nein": 1 2

#Gruppierung mittels cut()
(x = sample(20, size = 15, replace = TRUE))

## [1] 5 1 5 8 9 5 15 19 20 19 2 9 16 1 9

(f = cut(x, breaks = c(0, 10, 25, 50)))

## [1] (0,10] (0,10] (0,10] (0,10] (0,10] (0,10] (10,25] (10,25] (10,25] (10,25]
## [11] (0,10] (0,10] (10,25] (0,10] (0,10]
## Levels: (0,10] (10,25] (25,50]
```

Aktuelles Thema

1. Erste Schritte in R

2. Datenstrukturen in R

3. Schritt für Schritt mehr R

Daten: Import, Weiterverarbeitung und Export in R

Fehlende Werte in R

Grafiken in R

4. Erste Statistiken mit R

Daten laden und speichern

- ▶ Workspace bzw. einzelne Objekte davon speichern bzw. laden mittels

```
#kompletter Workspace in aktuelles working directory speichern  
save.image(file = "myworkspace.RData")  
  
#spezielle Objekte (diese stehen in mylist)  
save(list = mylist ,file="myfile.RData")  
  
#genau ein Objekt (hier mydata)  
save(mydata, file = "mydata.RData");  
  
#Daten / Workspace wieder laden  
load("mydata.RData")
```

- ▶ Textdateien (.txt oder .dat) mittels `read.table()`
- ▶ CSV-Dateien (.csv) mittels `read.csv()`
- ▶ SPSS-Dateien (.sav) mittels Foreign Package: `read.spss()`
- ▶ Daten speichern mittels `write()`

Auswahl von Teilstichproben

- ▶ in SPSS Auswahl von Teilmengen mittels Filter
- ▶ Zugriff auf Elemente in matrix / array mittels [Elementevektor]
- ▶ logische Operatoren (Vergleiche $A < B$, $A \leq B$, $A == B$, und: $A \& B$, oder: $A|B$, Negation $!A$, exclusives oder: $\text{xor}(A,B)$)
- ▶ R-Funktionen

```
subset (); subset.data.frame ()
```

▶ Beispiele

```
head(mtcars, n = 3)

##           mpg  cyl  disp  hp  drat    wt    qsec  vs  am  gear  carb
## Mazda RX4      21.0   6  160  110 3.90  2.620  16.46  0  1     4     4
## Mazda RX4 Wag  21.0   6  160  110 3.90  2.875  17.02  0  1     4     4
## Datsun 710     22.8   4  108   93 3.85  2.320  18.61  1  1     4     1

#Auswahl aller Autos aus mtcars mit Automatikgetriebe (am = 0)
mtcars.automatic = subset.data.frame(mtcars, subset = (mtcars$am==0))
```

(NA \triangleq “not available”)

```
#simulieren 5 normalverteilte Daten und hängen in x einen NA Wert an
nv = rnorm(n=5, mean=100, sd=4)
(x = c(nv, NA))

## [1] 104.5808 100.3318 94.7818 102.3856 109.8236 NA

mean(x); mean(x, na.rm=TRUE)

## [1] NA
## [1] 102.3807

#1. ueberpruefen, ob NA Werte und 2. wie viele fehlende bzw. gültige Werte
is.na(x); sum(is.na(x)); sum(!is.na(x))

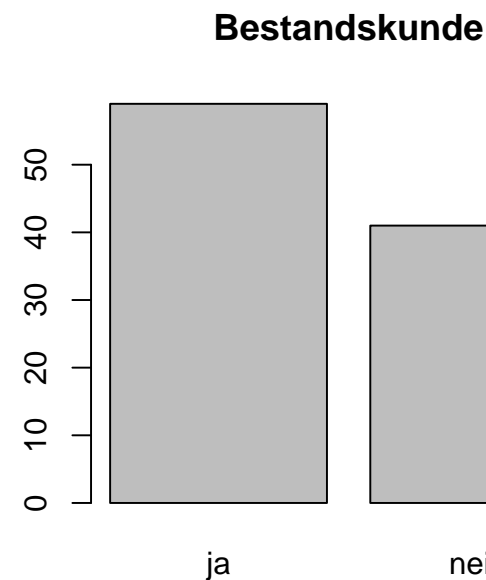
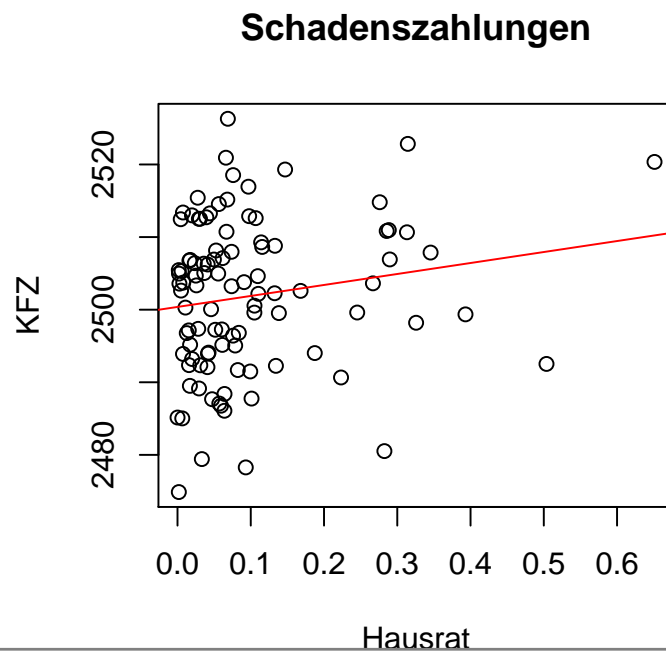
## [1] FALSE FALSE FALSE FALSE FALSE TRUE
## [1] 1
## [1] 5

(m = sum(x[!is.na(x)]) / sum(!is.na(x)))

## [1] 102.3807
```

```
plot(x); plot(x,y); plot(faktor); hist(x); curve(); boxplot();  
#hilfe zu grafikparametern  
?par
```

```
#Beispiele  
plot(schaden_hausrat, schaden_kfz, xlab="Hausrat", ylab="KFZ",  
      main="Schadenszahlungen")  
abline(lm(schaden_kfz~schaden_hausrat), col="red")  
plot(bestand, main="Bestandskunde")
```



Aktuelles Thema

1. Erste Schritte in R
2. Datenstrukturen in R
3. Schritt für Schritt mehr R
- 4. Erste Statistiken mit R**
Statistische Verteilungen

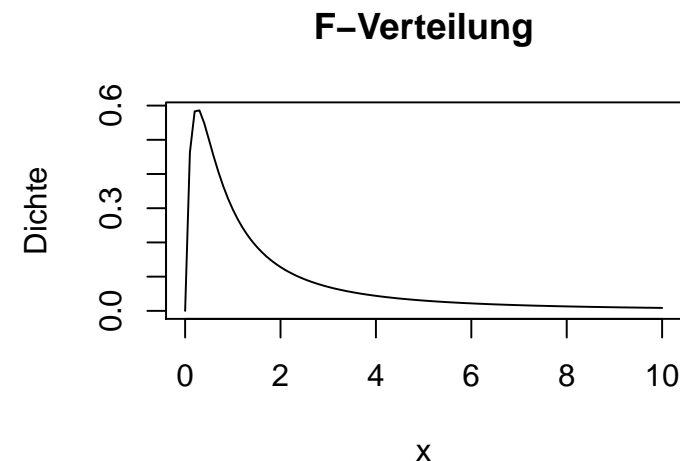
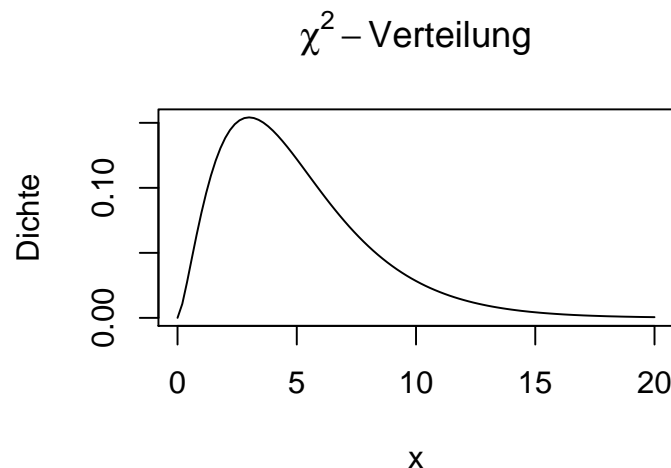
```

#rVERTEILUNG ... simuliere, dVERTEILUNG ... Dichte,
#pVERTEILUNG ... Verteilungsfkt, qVERTEILUNG ... Quantile
?Normal
rnorm(n=10, mean=100, sd=4)
dnorm(x=0, mean=0, sd=1)
pnorm(1.663, mean=0, sd=1)

#weitere Verteilungen
?Distributions
  
```

```

curve(dchisq(x, df=5), 0, 20, ylab="Dichte", main=expression(chi^2-Verteilung))
curve(df(x, df1=4, df2=2), 0, 10, ylab="Dichte", main="F-Verteilung")
  
```



Hilfreiche Links

- ▶ <http://personality-project.org/r/r.guide.html>
- ▶ <http://www.rseek.org/>
- ▶ <http://search.r-project.org/>
- ▶ http://de.wikibooks.org/wiki/GNU_R
- ▶ <http://www.r-stutorials.de/home>
- ▶ https://www.uni-muenster.de/imperia/md/content/psyifp/ae_holling/r-fehlermeldungen.pdf
- ▶ <http://r4ds.had.co.nz/>